Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-23 (2024) 25–30

Non-Negative Universal Differential Equations With Applications in Systems Biology ⋆

Maren Philipps[1] * **Antonia Körner** * **Jakob Vanhoefer** *
Dilan Pathirana^[2] * **Jan Hasenauer**^[3] *,**

* *Faculty of Mathematics and Natural Sciences, and the Life and Medical Sciences Institute (LIMES), Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany*
** *Computational Health Center, Helmholtz Zentrum München Deutsches Forschungszentrum für Gesundheit und Umwelt (GmbH), Neuherberg, Germany*

Abstract: Universal differential equations (UDEs) leverage the respective advantages of mechanistic models and artificial neural networks and combine them into one dynamic model. However, these hybrid models can suffer from unrealistic solutions, such as negative values for biochemical quantities. We present non-negative UDE (nUDEs), a constrained UDE variant that guarantees non-negative values. Furthermore, we explore regularisation techniques to improve generalisation and interpretability of UDEs.

Keywords: Kinetic modelling and control of biological systems, Parameter and state estimation, Parametric optimization, Neural networks, Systems biology.

1. INTRODUCTION

Systems biology aims to find a mechanistic understanding of biological processes and mathematical models are an important tool to achieve this. Mechanistic models are constructed to be consistent with the available information; however, their construction is limited by the current mechanistic understanding of the biological process. In practice, incomplete information is akin to multiple hypotheses about a process, which can be addressed by model selection. However, model selection can be time-consuming, particularly when there are large knowledge gaps.

A novel approach to address incomplete information is universal differential equations (UDEs), which combine dynamical mechanistic and machine learning (ML) models. UDEs represent known mechanisms explicitly, and unknown mechanisms by universal approximators like artificial neural networks (ANNs). These hybrid models have been shown to require less training data and improve interpretability over purely data-driven ML (Karniadakis et al., 2021). An open issue is that UDEs can produce negative values, even for strictly non-negative quantities such as molecular concentrations or population sizes. Dynamical mechanistic models, such as ordinary differential equations (ODEs), do not have this issue because the mechanisms can be chosen to ensure non-negativity.

Here, we present an extension to the UDE framework that ensures non-negativity (nUDE). We provide a proof of the non-negativity, and evaluate nUDEs on a synthetic and a real-world example. Moreover, we introduce a new regularisation method to control the over-fitting of the ANN in (n)UDEs. We find that our non-negativity approach may bias the ANN training; however, this bias can be reduced, and calibration efficiency and model quality can be preserved, by choosing the non-negativity factor carefully.

2. MODELLING

2.1 Mechanistic modelling with ODEs

Mechanistic modelling is facilitated by the conversion of domain knowledge into actionable mathematical expressions, which can be used to understand the modelled behaviour in a virtual setting. A significant benefit over non-mechanistic modelling is the ability to predict behaviour that is not represented in the available training data. However, model construction can be time-consuming. As the exemplary models in this work are taken from the literature, this process is not further discussed here. Reviews of this process are available in the literature, e.g. Villaverde et al. (2022).

In systems biology, ODEs are commonly used to describe the time-dependent rate-of-change of biological entities,

⋆ This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC 2047—390685813, EXC 2151—390873048) and under the project ID 432325352 – SFB 1454, by the German Federal Ministry of Education and Research (BMBF) under the CompLS program (GENImmune, grant no 031L0292F), and by the University of Bonn (via the Schlegel Professorship of J.H.).
Supplemental Material is available at `https://github.com/m-philipps/nUDE`.
^ *Joint senior authors.*
{[1]maren.philipps, [2]dilan.pathirana, [3]jan.hasenauer}@uni-bonn.de

10.1016/j.ifacol.2024.10.005

such as proteins on the level of cells, or groups of individuals on the level of populations. Here, we consider a general form of ODEs with initial conditions, i.e., initial value problems (IVPs):

$$\left.\begin{aligned}\frac{d\mathbf{x}(t,\boldsymbol{\theta}_{\mathrm{M}})}{dt} &= \mathbf{f}(\mathbf{x}(t,\boldsymbol{\theta}_{\mathrm{M}}),t,\boldsymbol{\theta}_{\mathrm{M}}),\\ \mathbf{x}(t_0,\boldsymbol{\theta}_{\mathrm{M}}) &= \mathbf{x_0}(\boldsymbol{\theta}_{\mathrm{M}}).\end{aligned}\right\} \tag{1}$$

The system changes with time according to the vector field $\mathbf{f} : \mathbb{R}^{n_\mathbf{x}} \times \mathbb{R}^{n_{\boldsymbol{\theta}_\mathrm{M}}} \to \mathbb{R}^{n_\mathbf{x}}$. The initial position, at $t = t_0$, is the initial condition $\mathbf{x_0} : \mathbb{R}^{n_{\boldsymbol{\theta}_\mathrm{M}}} \to \mathbb{R}^{n_\mathbf{x}}$. The model is parameterised by $\boldsymbol{\theta}_{\mathrm{M}}$, which can contain values such as population growth rate constants.

2.2 Machine learning with ANNs

In supervised machine learning, input-output pairs are used to train an unknown function that represents the input-output mapping. Some ANNs, such as multi-layer feedforward networks, are universal approximators in the limit case, meaning they are capable of approximating functions to arbitrary precision (see e.g. Hornik et al. (1989)). These ANNs are structured into layers of *neurons*, where each neuron ψ of each layer $\boldsymbol{\phi}$ is the composition of one affine and (usually) one non-linear transformation function. The fully-connected ANN $\mathbf{U}$ can then be understood as a composition of layers, i.e.

$$\mathbf{U}(\boldsymbol{\nu},\boldsymbol{\theta}_{\mathrm{U}}) = \left(\boldsymbol{\phi}_L \circ \boldsymbol{\phi}_{L-1} \circ ... \circ \boldsymbol{\phi}_1\right)(\boldsymbol{\nu},\boldsymbol{\theta}_{\mathrm{U}}),$$

where $\boldsymbol{\theta}_{\mathrm{U}}$ are the weight and bias parameters of the affine functions, $\boldsymbol{\nu} \in \mathbb{R}^{n_\boldsymbol{\nu}}$ is the input, and L is the total number of layers (Gouk et al., 2021).

Many commonly used activation functions are Lipschitz continuous (Gouk et al., 2021). These include the hyperbolic tangent $\tanh(z)$, the logistic sigmoid $(1+\exp(-z))^{-1}$, and the rectified linear unit (ReLU) $\max(0,z)$. As ANNs with these activation functions are thereby compositions of Lipschitz-continuous functions, these ANNs are also Lipschitz continuous; a property that we use to prove Theorem 1.

We describe the architecture of these ANNs by the number of neurons in each layer. For example, 3/3/2 is a fully-connected feedforward ANN with 3 neurons in the first and second layers, and 2 neurons in the output layer. Each neuron i in layer l outputs $\psi_{l,i} = \sum_j \mathcal{A}_{l,i,j}(w_{l,i,j}\psi_{l-1,j} + b_{l,i,j})$, where j is the neuron index in the previous layer, $\psi_{l-1,j}$ is the output from neuron j in the previous layer $j-1$, and $\mathcal{A}_{l,i,j}$, $w_{l,i,j}$ and $l_{k,i,j}$ are the activation function, weight and bias, respectively. The output from layer l with n neurons is then $\boldsymbol{\phi}_l = (\psi_{l,1}, \cdots, \psi_{l,n})$.

2.3 Universal differential equations

Neural ODEs are ODEs similar to (1), but with an ANN as their right-hand-side, i.e.

$$\begin{aligned}\frac{d\mathbf{x}(t,\boldsymbol{\theta}_{\mathrm{U}})}{dt} &= \mathbf{U}(\boldsymbol{\nu}(\mathbf{x}(t,\boldsymbol{\theta}_{\mathrm{U}})),\boldsymbol{\theta}_{\mathrm{U}}),\\ \mathbf{x}(t_0,\boldsymbol{\theta}_{\mathrm{U}}) &= \mathbf{x_0}(\boldsymbol{\theta}_{\mathrm{U}}),\end{aligned}$$

where the input $\boldsymbol{\nu}(\mathbf{x}(t,\boldsymbol{\theta}_{\mathrm{U}}))$ is now some function of the state $\mathbf{x}$.

While mechanistic modelling and neural ODEs with ANNs both have important use cases, both approaches have drawbacks. UDEs have been introduced to exploit the strengths of each approach, and to enable modelling of partially unknown biological processes (Oliveira, 2004). A general form for UDEs in different modelling formalisms is given in Rackauckas et al. (2020). In the ODE context, a formulation for UDEs is given by

$$\left.\begin{aligned}\frac{d\mathbf{x}(t,\boldsymbol{\theta})}{dt} &= \mathbf{f}(\mathbf{x}(t,\boldsymbol{\theta}),t,\boldsymbol{\theta}_{\mathrm{M}}) + \mathbf{U}(\boldsymbol{\nu}(\mathbf{x}(t,\boldsymbol{\theta})),\boldsymbol{\theta}_{\mathrm{U}}),\\ \mathbf{x}(t_0,\boldsymbol{\theta}) &= \mathbf{x_0}(\boldsymbol{\theta}),\end{aligned}\right\} \tag{2}$$

where $\mathbf{U}$ enables modelling unknown process mechanisms in addition to the known mechanisms $\mathbf{f}$. Here, $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathrm{M}}, \boldsymbol{\theta}_{\mathrm{U}})$, with mechanistic parameters $\boldsymbol{\theta}_{\mathrm{M}}$. Although $\mathbf{U}$ can be any universal approximator, in this study we only consider fully-connected feedforward ANNs.

We note that the system in (2) is not a universal approximator for a dynamical system, despite $\mathbf{U}$ being a universal approximator. However, this can be achieved by adding state variables to the system that are equipped with dynamics that are wholly-modelled in terms of a universal approximator (Dupont et al., 2019).

2.4 Maximum likelihood estimation

Mechanistic models, ANNs, and UDEs often contain unknown parameters, which can be estimated from data. One approach is to find the maximum likelihood estimate (MLE), which is the choice of parameter values $\boldsymbol{\theta}_{MLE}$ that maximises the probability of observing measurements $\bar{\mathbf{y}}$, i.e., the likelihood of the data under some system parameterised by $\boldsymbol{\theta}$. This requires an observation model $\mathbf{h} : \mathbb{R}^{n_\mathbf{x}} \to \mathbb{R}^{n_\mathbf{y}}$ that maps model state space to data observation space. In general, measurements in the biological sciences are significantly noisy, hence the observables $\mathbf{y} = \mathbf{h}(\mathbf{x}(t,\boldsymbol{\theta}),\boldsymbol{\theta})$ are related to the data by $\bar{y}_{t_i,y_i} = y_{t_i,y_i} + \epsilon_{t_i,y_i}$, where $t_i \in 1,...,n_t$ and $y_i \in 1,...,n_\mathbf{y}$ are used to index over measurements by timepoint and observable, respectively, and ϵ_{t_i,y_i} is measurement-specific noise.

For numerical efficiency, we minimise the negative log-likelihood function,

$$J(\boldsymbol{\theta}) = \frac{1}{2}\sum_{t_i,y_i}\log(2\pi\sigma^2_{t_i,y_i}) + \frac{(\bar{y}_{t_i,y_i} - y_{t_i,y_i}(\boldsymbol{\theta}))^2}{\sigma^2_{t_i,y_i}},$$

for i.i.d. Gaussian noise, i.e. $\epsilon_{t_i,y_i} \sim N(0,\sigma^2_{t_i,y_i})$.

3. TOWARDS BIOLOGICALLY MEANINGFUL UDES

A common issue in machine learning is over-fitting, characterised by the model adapting to noise or artefacts in the training data, which is deleterious for generalisation beyond the training domain (Ying, 2019). This issue is more prominent in ML compared to mechanistic models that are constrained by domain knowledge. A special case of over-fitting in UDEs is the absorption of the dynamics that are encoded in the mechanistic terms ($\mathbf{f}$ in system (2)). A broad spectrum of approaches to mitigate over-fitting have been introduced in ML, including early stopping, noise injection, and stochastic gradient descent. Few of these approaches have been transferred to the field of UDEs. One common approach that we tested here is to regularise the system during training by introducing a *learning bias* (Karniadakis et al., 2021).

Another limitation of UDEs is that their dynamics are not necessarily biologically meaningful. For example, a naïve function approximator would not adhere to the principles of mass conservation or non-negativity of biological quantities. Just as mechanistic models can be designed to implicitly comply with such fundamental principles, mathematical constraints for ANNs can be used as an *inductive bias* to strictly enforce biologically-reasonable model behaviour (Karniadakis et al., 2021).

We describe the learning bias *parameter regularisation*, introduce the learning bias *output regularisation*, and introduce the inductive bias *non-negative UDEs* (nUDEs).

3.1 Parameter regularisation:

Parameter regularisation aims to reduce the magnitude of the parameters as a proxy for model flexibility. In particular, ℓ_1 and ℓ_2 norms are frequently used to directly penalise model parameters (Goodfellow et al., 2016). Here, we use the ℓ_2 (also known as Euclidean) norm of $\boldsymbol{\theta}_{\mathrm{U}}$, $\|\boldsymbol{\theta}_{\mathrm{U}}\|_2 = \sqrt{\sum_i {\theta_{U_i}}^2}$, yielding the regularised objective

$$J(\boldsymbol{\theta}) + \lambda_p \|\boldsymbol{\theta}_{\mathrm{U}}\|_2^2,$$

with regularisation parameter $\lambda_p \geq 0$.

3.2 Output regularisation:

As parameter regularisation only indirectly limits the impact of $\mathbf{U}$ on the solution, we also consider a novel regularisation scheme, which we will refer to as output regularisation. We compute non-zero contributions of $\mathbf{U}$ to the solution directly with

$$R(\boldsymbol{\theta}) = \int_{t_0}^{t_f} \|\mathbf{U}_R(\boldsymbol{\theta})\|_2 \, \mathrm{d}t,$$

where $\mathbf{U}_R = \mathbf{U}$ and $\mathbf{U}_R = \mathbf{N} \odot \mathbf{U}$ in the UDE and nUDE (see Section 3.3) cases, respectively. We set t_f to the time of the last measurement in the training data and add the penalty to the regularised objective function as

$$J(\boldsymbol{\theta}) + \lambda_o R(\boldsymbol{\theta})^2,$$

with regularisation parameter $\lambda_o \geq 0$.

3.3 Non-negative UDEs

In this section, we present a formulation of a constrained UDE that ensures that state variables cannot become negative. We consider the model structure

$$\begin{aligned}\frac{d\mathbf{x}_{\mathrm{nUDE}}(t,\boldsymbol{\theta})}{dt} &= \mathbf{f}(\mathbf{x}_{\mathrm{nUDE}}(t,\boldsymbol{\theta}), t, \boldsymbol{\theta}_{\mathrm{M}}) \\ &\quad + \mathbf{N}(\mathbf{x}_{\mathrm{nUDE}}(t,\boldsymbol{\theta})) \odot \mathbf{U}(\boldsymbol{\nu}(\mathbf{x}_{\mathrm{nUDE}}(t,\boldsymbol{\theta})), \boldsymbol{\theta}_{\mathrm{U}}),\end{aligned} \tag{3}$$

with Lipschitz-continuous functions $\mathbf{U} : \mathbb{R}^{n_\nu} \to \mathbb{R}^{n_\varkappa}$ and $\mathbf{N} : \mathbb{R}^{n_\varkappa} \to \mathbb{R}^{n_\varkappa}$, and choosing $\mathbf{N} : \lim_{x_i \to 0} N_i(\mathbf{x}) = 0 \, \forall i \in \{1, \ldots, n_{\mathbf{x}}\}$ (e.g., $\mathbf{N}(\mathbf{x}) = \mathbf{x}$). $\odot$ is the element-wise (Hadamard) product. In the following, we present the properties of this non-negative UDE (nUDE). Some function inputs are omitted for brevity after their first use.

Theorem 1. Consider the ODEs IVP,

$$\begin{aligned}\frac{d\mathbf{x}_{\mathrm{ODE}}(t,\boldsymbol{\theta}_{\mathrm{M}})}{dt} &= \mathbf{f}(\mathbf{x}_{\mathrm{ODE}}(t,\boldsymbol{\theta}_{\mathrm{M}}), t, \boldsymbol{\theta}_{\mathrm{M}}), \\ \mathbf{x}_{\mathrm{ODE}}(t_0,\boldsymbol{\theta}_{\mathrm{M}}) &\geq \mathbf{0},\end{aligned}$$

where $\mathbf{f}$ is Lipschitz-continuous. If the non-negative quadrant is invariant under $\mathbf{f}$, i.e. $f_i|_{x_i=0} \geq 0 \, \forall i \in \{1, \ldots, n_{\mathbf{x}}\}$, then $x_{nUDE,i} \geq 0 \, \forall t \geq t_0$ is a nUDE system (3) with the same $\mathbf{f}$.

Proof: As the right-hand-side of (3) is composed of Lipschitz-continuous functions, the nUDE IVP has a unique solution $\mathbf{x}_{\mathrm{nUDE}}(t, \boldsymbol{\theta})$, by the Picard–Lindelöf theorem. The initial value $\mathbf{x}_{\mathrm{nUDE}}(t_0, \boldsymbol{\theta})$ is non-negative. We will show that the solution remains non-negative, with a proof by contradiction.

Assume there exists some $i \in 1, ..., n_{\mathbf{x}}$ and $\tilde{t} > t_0$ s.t. $x_{\mathrm{nUDE},i}(\tilde{t}, \boldsymbol{\theta}) < 0$. As the initial condition is non-negative, there must be some $t^* \in [t_0, \tilde{t}]$ s.t. $x_{\mathrm{nUDE},i}(t^*, \boldsymbol{\theta}) = 0$ and its derivative

$$(f_i + N_i U_i)|_{t=t^*} < 0. \tag{4}$$

As $\mathbf{U}$ and $\mathbf{N}$ are continuous $U_{\max} := \max_{t \in [t_0, t^*]} U_i < \infty$ and $N_i|_{x_{\mathrm{nUDE},i}=0} = 0$. Hence, $|\mathbf{N} \odot \mathbf{U}|_i = |N_i \cdot U_i| \leq |N_i \cdot U_{\max}| = 0$ at $t = t^*$, and (4) simplifies to $f_i|_{t=t^*} < 0$. However, given $f_i|_{x_i=0} \geq 0$, we arrive at the contradiction $f_i|_{t=t^*} \geq 0$, hence no such $\tilde{t}$ exists. □

4. IMPLEMENTATION AND BENCHMARKING

4.1 Implementation

We implemented simulation and training of (n)UDEs using established software packages. Simulation, objective function evaluation and gradient calculation was implemented in the Advanced Multi-language Interface for CVODES and IDAS (AMICI) (Fröhlich et al., 2021). To ensure scalability, adjoint sensitivity analysis was employed (Fröhlich et al., 2017). Parameter estimation problems were specified using the Parameter Estimation Table (PEtab) format (Schmiester et al., 2021). We work with two pre-existing biologically-inspired models (termed Lotka-Volterra and Boehm), with details provided in the following sections.

As suggested in the literature (Hass et al., 2019), we $\log_{10}$-transform the mechanistic parameters $\boldsymbol{\theta}_{\mathrm{M}}$ for estimation. The ANN parameters $\boldsymbol{\theta}_{\mathrm{U}}$ are not transformed. $\boldsymbol{\theta}_{\mathrm{U}}$ were generally initialised to be small values and estimated $\in [-10, 10]^{n_{\boldsymbol{\theta}_{\mathrm{U}}}}$. We used multi-start (1000 starts), gradient-based optimisation with the Fides optimiser (Fröhlich and Sorger, 2022) via the Python Parameter Estimation Toolbox (pyPESTO) (Schälte et al., 2023). We initialised the starts by drawing 1000 samples of $\boldsymbol{\theta}_{\mathrm{M}}$ and $\boldsymbol{\theta}_{\mathrm{U}}$, and we reused these 1000 sets of vectors across all comparable experiments, i.e., when using the same model (Lotka-Volterra or Boehm) and ANN architecture.

When assessing whether a solution is negative, we define "negative" to be when any state variable in the solution drops below a small negative value, which was chosen to filter for numerical noise. This was -1e-7 for the Lotka-Volterra model and -1e-13 for the Boehm model, which are within one order of magnitude of the absolute simulation tolerances used with each problem.

4.2 Lotka-Volterra model

For the demonstration of UDEs, nUDES and the different regularisations, we considered a Lotka-Volterra system

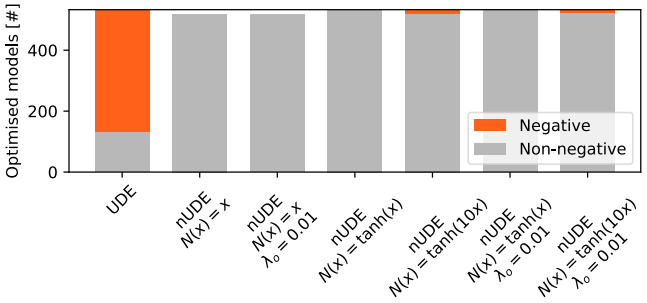

Fig. 1. Solution characteristics for the Lotka-Volterra model: the number of optimisation runs yielding solutions with negative (orange) or strictly non-negative (gray) values for the predator or prey abundances.

describing predator-prey population dynamics. The abundance of prey and predator are denoted by x_1 and x_2, respectively. The standard Lotka-Volterra ODE system is

$$\left.\begin{aligned}\frac{dx_1}{dt} &= \alpha x_1 - \beta x_1 x_2,\\ \frac{dx_2}{dt} &= \delta x_1 x_2 - \gamma x_2,\end{aligned}\right\} \tag{5}$$

with mechanistic parameters $\boldsymbol{\theta}_{\mathrm{M}} = (\alpha, \beta, \gamma, \delta)$.

As a test case for modelling unknown mechanisms, we consider that the interaction terms of the system are unknown and replace them with an ANN $\mathbf{U} = (U_1, U_2)$. This yields the UDE system ($N_1 = N_2 = 1$) and nUDE system

$$\left.\begin{aligned}\frac{dx_1}{dt} &= \alpha x_1 + N_1(x_1)U_1(\boldsymbol{\nu}, \boldsymbol{\theta}_{\mathrm{U}}),\\ \frac{dx_2}{dt} &= N_2(x_2)U_2(\boldsymbol{\nu}, \boldsymbol{\theta}_{\mathrm{U}}) - \gamma x_2.\end{aligned}\right\} \tag{6}$$

We chose a simple 2/2 ANN for $\mathbf{U}$ with tanh activation functions in the first layer, and the identity in the output layer. The input $\boldsymbol{\nu} := \mathbf{x}$ is the state vector.

We initialised each entry of $\boldsymbol{\theta}_{\mathrm{M}}$ (α and γ) randomly in $[10^{-3}, 10^{3}]$ (uniform distribution), and constrained them by the same bounds during estimation. Approximately 50% of all start points could not be simulated and optimised, for example due to exponential blow-up.

Synthetic data for training and validation were generated by simulating the system in (5) with 100 time units with $\alpha = 1.3$, $\beta = 0.9$, $\gamma = 0.8$, and $\delta = 1.8$, and $\mathbf{x}(t = 0) = (0.44249296, 4.6280594)$. The first 20 time units of simulated data were used for training, and the next 80 time units for validation. The training data had 15% multiplicative noise ($\mathcal{N}(\mathbf{0}, 0.15\mathbf{x})$) added to it, to represent noise-corrupted data.

We simulated each optimisation result to check for non-negativity (Fig. 1). More than half of the UDE fits produced negative populations. All nUDE models had zero negative populations, except with "$N(x) = \tanh(10x)$". The ODE solver may have numerical issues near zero due to the larger second derivative of $\tanh(10x)$, than $\tanh(x)$.

The "nUDE; $N(x) = x$; $\lambda_o = 0.01$" case performed best on the training data (not shown), and on the validation data, and the "nUDE; $N(x) = x$" was next best on validation

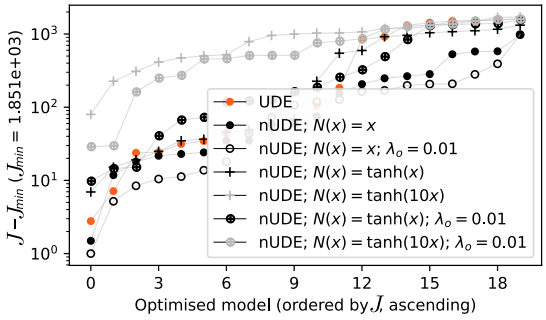

Fig. 2. Waterfall plot for the Lotka-Volterra model. The objective function value J on the validation data are shown for the 20 best fits on the training data.

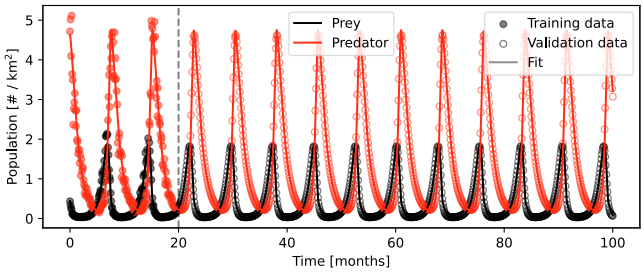

Fig. 3. Fit and prediction for the Lotka-Volterra UDE. The vertical dashed line indicates the training and validation data split. The fits and predictions from the "nUDE; $N(x) = x$" and "nUDE; $N(x) = x$; $\lambda_o = 0.01$" models are visually indistinguishable to the UDE, and are not shown.

data (Fig. 2). However, the best UDE fit was also very good (Fig. 3).

4.3 Boehm model

As a second example we consider the Boehm model (Boehm et al., 2014), which describes the STAT5 dimerisation process. It is fully specified by eight ODEs and nine estimated parameters. The measurements are mapped to the eight state variables through a nonlinear observation function. Implementation and training details are specified in the supplemental materials section 2. To evaluate UDEs and nUDEs, we consider two scenarios for the Boehm model that differ in the effect that the ANN component can have on the overall dynamics:

Scenario 1: Like in the synthetic Lotka Volterra example, we assume that one mechanism is unknown, here the export and dimer dissociation of `nucpApA`, and remove the term from the ODE. Instead, we introduced a 5/5/5/2 ANN with 82 weight and bias parameters $\boldsymbol{\theta}_{\mathrm{U}}$. This ANN takes only a subset of the state vector as input (specifically, the `nucpApA` species) and modifies the dynamics of two species (`nucpApA` and `STAT5A`).

Scenario 2: A more flexible ANN component is used to emphasise the effect of regularisation. This 5/5/5/5 ANN has the same dimensions in the hidden layers as in *Scenario 1* but takes three state variables as inputs, and modifies the dynamics of five species, which increases the size of $\boldsymbol{\theta}_{\mathrm{U}}$ to 110 free parameters. This scenario represents a greater degree of uncertainty about the missing mechanisms in the

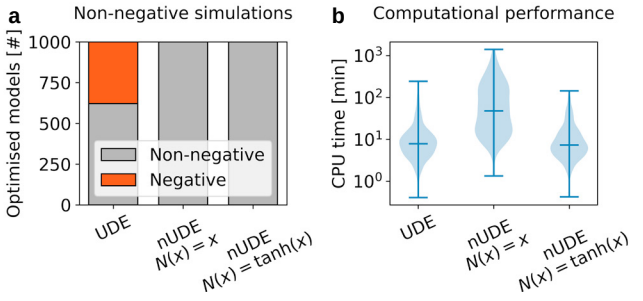

Fig. 4. Boehm *Scenario 1* comparison between UDE and nUDEs. a) Amount of models that stayed non-negative in their trajectories, and b) distribution of model calibration times per method. Horizontal lines indicate the minimum, medium and maximum.

model, because the ANN can affect more state variables directly.

We first consider *Scenario 1* to assess the non-negativity constraint in a realistic setting. Of the 1000 starts, 623 optimised UDE fits had non-negative values, while all 1000 nUDE fits were non-negative (Fig. 4a). However, we found that the overall computation time for optimisation was significantly increased when training the nUDEs with $\mathbf{N}(\mathbf{x}) = \mathbf{x}$ (Fig. 4b). We saw an increased number of optimiser iterations, and simulation time (Supplemental Fig. 3) and furthermore observed predominantly non-smooth trajectories among the best ($\mathbf{N}(\mathbf{x}) = \mathbf{x}$)-nUDE results, indicative of over-fitting (Fig. 5a). When using $\mathbf{N}(\mathbf{x}) = \tanh(\mathbf{x})$ however, all 1000 parameterised nUDEs stayed non-negative in their trajectories (Fig. 4a) while the computational cost (Fig. 4b) and the UDE's quality of fit (Fig. 5a) were recovered. The 2% of best fits shown in Fig. 5 agree much better for the ($\mathbf{N}(\mathbf{x}) = \tanh(\mathbf{x})$)-nUDE than in the ($\mathbf{N}(\mathbf{x}) = \mathbf{x}$)-nUDE, indicating better convergence.

We used the *Scenario 2* UDE variant with a larger ANN component to assess the effect of regularisation. The ANN flexibility has a considerable effect on UDE convergence and over-fitting, as shown by the difference in trajectories between the unregularised *Scenario 1* and *Scenario 2* UDEs (Fig. 5a and 5b). With increasing ANN complexity the number of UDEs with negative values increased from 37.7% in *Scenario 1* to 87.7% in *Scenario 2*.

There is a substantial difference in the quality of fits between the unregularised and regularised UDEs, as apparent from the best 2% of fits (Fig. 5b). The unregularised UDEs tend to over-fit the training data, characterised by a tight fit to the measurements and a high variability in their trajectories between measurements, with frequent spikes. The regularised UDEs on the other hand have a high agreement between the 20 best models and produce smooth trajectories, as shown for the parameter regularisation in Fig. 5b. We observed similar trends between parameter and output regularisation.

The UDE is expected to fit measurements more closely than the fully mechanistic model due to the flexibility of ANNs. However, this does not directly indicate its generalisation capacity in predictions or inference of non-observed states variables. In the real-world Boehm example, the reference for non-observed state variables is not the true solution, which is unknown, but the optimal solution from

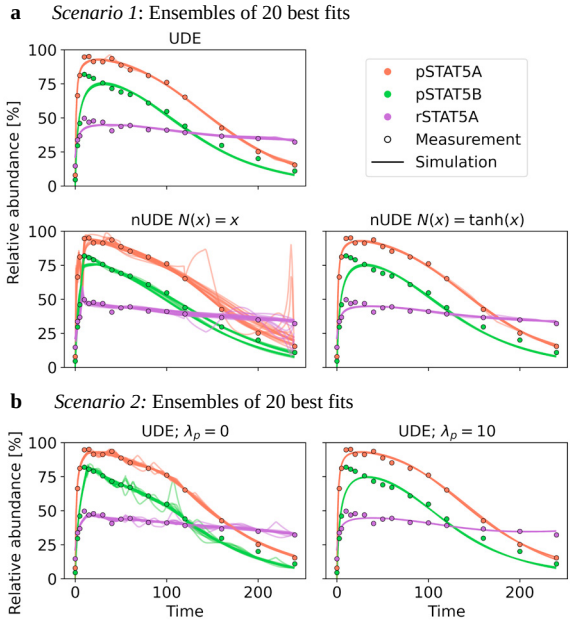

Fig. 5. Ensembles of 20 best UDE model variants. **a)** *Scenario 1*: ANN with 1 input/2 outputs. Best fits are shown for UDE, nUDE ($\mathbf{N}(\mathbf{x}) = \mathbf{x}$) and nUDE ($\mathbf{N}(\mathbf{x}) = \tanh(\mathbf{x})$), no regularisation. **b)** *Scenario 2*: ANN with 3 inputs/5 outputs. Best fits are shown for the unregularised UDE and a parameter-regularised UDE with $\lambda_p = 10$.

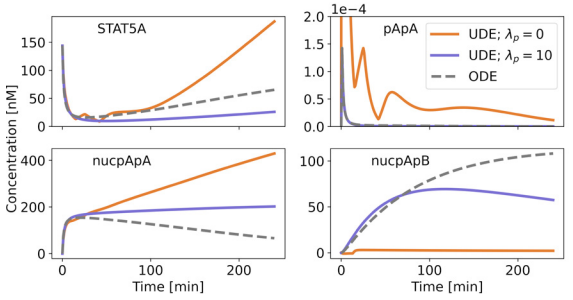

Fig. 6. Boehm *Scenario 2*: Simulation of best unregularised UDE (orange) and UDE with parameter regularisation ($\lambda_p = 10$, purple). Shown are the simulations for 4/8 state variables.

the fully mechanistic ODE model. Compared to the ODE reference, the normalised mean squared error (NMSE) for the best UDE was significantly improved by parameter and output regularisation (NMSE with $\lambda_p = \lambda_o = 0$: 145.99, $\lambda_p = 10$: 12.83, $\lambda_o = 0.1$: 12.69). While it's unclear whether these trajectories are correct, they are more in line with the previously published reference. Therefore, some regularisation is expected to reduce over-fitting and recover biologically reasonable and interpretable non-observed state variables. This advantage is evident in simulations, where regularisation mitigates blow-up and strong fluctuations observed in the unregularised UDE (Fig. 6).

5. DISCUSSION

In this manuscript, we presented and evaluated different types of regularisation on the universal components of UDEs describing biological processes. In particular, we

introduced (i) an output regularisation to avoid overfitting, and (ii) a regularisation of the UDE structure to ensure non-negativity.

Our theoretical result guarantees that the solution is non-negative everywhere, up to numerical noise, and this constraint is generally applicable to biological modelling, where entities like molecular concentrations and population sizes are often non-negative. Our experimental results demonstrate that the non-negative constraint works in principle on a synthetic Lotka-Volterra example, and in practice on the real-life Boehm example.

The choice of $\mathbf{N}(\mathbf{x})$ can itself be mechanistically informed. If there is some prior knowledge that a missing mechanism affecting x_i has the factor x_i, then choosing $N_i(\mathbf{x}) = x_i$ may improve the learning problem for $\mathbf{U}$. However, if the missing mechanism does not depend on x_i, then $\mathbf{U}$ needs to counter this in addition to learning the missing mechanism. In such cases, we suggest the bounded $N_i(\mathbf{x}) = \tanh(\alpha x_i)$, which does not grow with x_i except near 0, according to α. This can have substantial computational benefits (Fig. 4). However, larger choices of α can increase numerical error (Fig. 1), so alternative choices of $\mathbf{N}$ are an important open topic.

We found that $\mathbf{N}$ did ensure non-negativity in computational experiments, but only up to numerical error. Tailored ODE solvers can ensure that user-provided constraints are satisfied by performing additional integration steps as a constraint is approached (Eich, 1993). This could be used to remove negativity due to numerical error in nUDEs, but does not resolve the negativity in standard UDEs.

In principle, our regularisation strategies are applicable to a variety of universal approximators, modelling formalisms such as partial differential equations, and choices of $\mathbf{N}$. We present some limited benchmarking here. Our results for the Boehm model in *Scenario 2* suggest that, as the amount of prior assumptions on the missing dynamics decreases, the user is forced to choose a more expressive ANN ($\mathbf{U}$), and the importance of regularisation increases. Commonly, ODE models in systems biology are characterised by a stoichiometric matrix and a flux vector, which can be exploited to encode further biological properties into ANNs like conservation of mass (Pinto et al., 2022). Our approach naturally extends to this setting and, moreover, can be used to specify directionality in the reactions. More comprehensive benchmarking is required to uncover best practices when modelling unknown mechanisms.

Hybrid models promise to bridge the gap between the interpretability of mechanistic models, and the predictive capabilities of machine learning models. Context-specific modelling choices can improve the performance of hybrid models substantially. We integrated UDEs into standard workflows for systems biology and showed that biologically-reasonable predictions are possible, without sacrificing computational efficiency.

ACKNOWLEDGEMENTS

We are grateful to Polina Lakrisenko for fruitful discussions. Optimisation was performed on the Bonna and Unicorn clusters at the University of Bonn.

REFERENCES

Boehm, M.E., Adlung, L., Schilling, M., et al. (2014). Identification of isoform-specific dynamics in phosphorylation-dependent stat5 dimerization by quantitative mass spectrometry and mathematical modeling. *J. Proteome Res.*, 13(12), 5685–5694.

Dupont, E., Doucet, A., and Teh, Y.W. (2019). Augmented neural odes. *Adv. Neur. In.*, 32.

Eich, E. (1993). Convergence Results for a Coordinate Projection Method Applied to Mechanical Systems with Algebraic Constraints. *SIAM J. Numer. Anal.*, 30(5), 1467–1482. doi:10.1137/0730076.

Fröhlich, F., Kaltenbacher, B., Theis, F.J., and Hasenauer, J. (2017). Scalable parameter estimation for genome-scale biochemical reaction networks. *PLOS Comput. Biol.*, 13(1), e1005331.

Fröhlich, F. and Sorger, P.K. (2022). Fides: Reliable trust-region optimization for parameter estimation of ordinary differential equation models. *PLOS Comput. Biol.*, 18(7), e1010322.

Fröhlich, F., Weindl, D., Schälte, Y., et al. (2021). AMICI: High-performance sensitivity analysis for large ordinary differential equation models. *Bioinformatics*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Gouk, H., Frank, E., Pfahringer, B., et al. (2021). Regularisation of neural networks by enforcing Lipschitz continuity. *Mach. Learn.*, 110(2), 393–416.

Hass, H., Loos, C., Raimúndez-Álvarez, E., et al. (2019). Benchmark problems for dynamic modeling of intracellular processes. *Bioinformatics*, 35(17), 3073–3082.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward fetworks are universal approximators. *Neural Netw.*, 2(5), 359–366.

Karniadakis, G.E., Kevrekidis, I.G., Lu, L., et al. (2021). Physics-informed machine learning. *Nat. Rev. Phys.*, 3(6), 422–440.

Oliveira, R. (2004). Combining first principles modelling and artificial neural networks: a general framework. *Comput. Chem. Eng.*, 28(5), 755–766.

Pinto, J., Mestre, M., Ramos, J., et al. (2022). A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Computers & Chemical Engineering*, 165, 107952.

Rackauckas, C., Ma, Y., Martensen, J., et al. (2020). Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385.

Schälte, Y., Fröhlich, F., Jost, P.J., et al. (2023). pyPESTO: A modular and scalable tool for parameter estimation for dynamic models. *Bioinformatics*, 39(11), btad711.

Schmiester, L., Schälte, Y., Bergmann, F.T., et al. (2021). PEtab—Interoperable specification of parameter estimation problems in systems biology. *PLOS Comput. Biol.*, 17(1), e1008646.

Villaverde, A.F., Pathirana, D., Fröhlich, F., et al. (2022). A protocol for dynamic model calibration. *Brief. Bioinform.*, 23(1), bbab387.

Ying, X. (2019). An overview of overfitting and its solutions. In *J. Phys. Conf. Ser.*, volume 1168, 022022.